ADVANCED CONTROL OF MEMS PROBING DEVICES

BY

XUEMENG ZHANG

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Adviser:

Professor Srinivasa M Salapaka

Abstract

This work is aimed at developing a control-system theoretic approach for addressing certain performance issues that arise in micro-electro-mechanical systems (MEMS). In particular, it focuses on applications such as nano-positioning, where control design becomes necessary to meet high resolution, bandwidth, and reliability (robustness) demands especially when there is significant model uncertainty and instrumentation noise. In this article, a systematic control design from robust control approach is demonstrated on a micro probing device with electrically separated sensing combs and driving combs. The system is identified through experimental input-output data and the hardware is setup in such a way that the resulting model is a linear time-invariant model with appropriate choice of variables even when the the underlying constitutive laws are nonlinear. Controllers are developed based on PID and H_∞ control design methodologies. Control algorithms from PID control and robust control have been implemented on dSpace digital processing platform. The implemented control (H_∞) design demonstrates a significant ($\approx 400\%$) improvement in the bandwidth, where the bandwidths from the closed-loop sensitivity and complementary-sensitivity functions respectively are 68 Hz and 74 Hz. A significant improvement in reliability and repeatability (robustness to uncertainties) as well as noise attenuation is also demonstrated through this design.

To mom and dad

Acknowledgements

I would like to take this opportunity to thank my advisor, Dr. Srinivasa Salapaka, for his academical help and encouragement. Thanks also go to Dr. Placid Ferriera and Dr. Jingyan Dong for their valuable advice. I also appreciate the help from my fellow student, Bonjin Koo, for his fabrication work on which I could implement my controllers.

I also want to express my sincere gratitude towards my labmates, Dr. Chibum Lee and Gayathri Mohan, for their help from start throughout my stay in University of Illinois. I would also like to thank my friends I met here or has been acquainted long before, for their friendship which makes me cherish my life more.

Last but not the least, I owe so much to my parents, Yihou Zhang and Guiying Liu, for their longtime love. I will take the opportunity here to send my special thanks to them.

Table of Contents

List of Figures

Chapter 1

Introduction

In Micro-Electro-Mechanical-System (MEMS) devices, capacitive structures which consist of electrostatic combs are being widely used, especially in MEMS probing devices, since it was first introduced by Tang [1]. As an actuator, electrostatic comb drive has several advantages such as simplifying actuating process and being easy to manufacture, compared to others such as piezoelectric material and shape memory alloy. Moreover, electrostatic comb drive has a faster response and it is easy to implement a feedback control since the driving force does not depend on actuator position but only on the applied voltage.

For applications in fast nano-positioning or nano-scanning, accurate displacement monitoring of comb drive is needed. This requires a high-resolution sensing scheme. The displacement or corresponding capacitance (less than 1 pF) in electrostatic comb drives is usually very small, which adds difficulty to the sensing module design. An optical sensor based on laser diode was implemented by Borovic [2] which decoupled the sensing from actuation, however it gives a limited operating range since the sensor has a linear measurement only within limited range. A simultaneous actuation and displacement sensing scheme was successfully realized by Dong [3] but it requires a low pass filter in its scheme and thus seriously limited its bandwidth. Other sensing schemes such as microscope [4] and AFM have also been used to detect the displacement of comb drives, but none of them could be possibly used in a high bandwidth feedback control. In addition several on-chip capacitance measurement schemes are reported in [5], [6], which achieve high resolution measurements and show promise for functioning as sensors in the closed loop system of comb drives.

In this work, the comb drive is designed with control design in mind. In most devices in

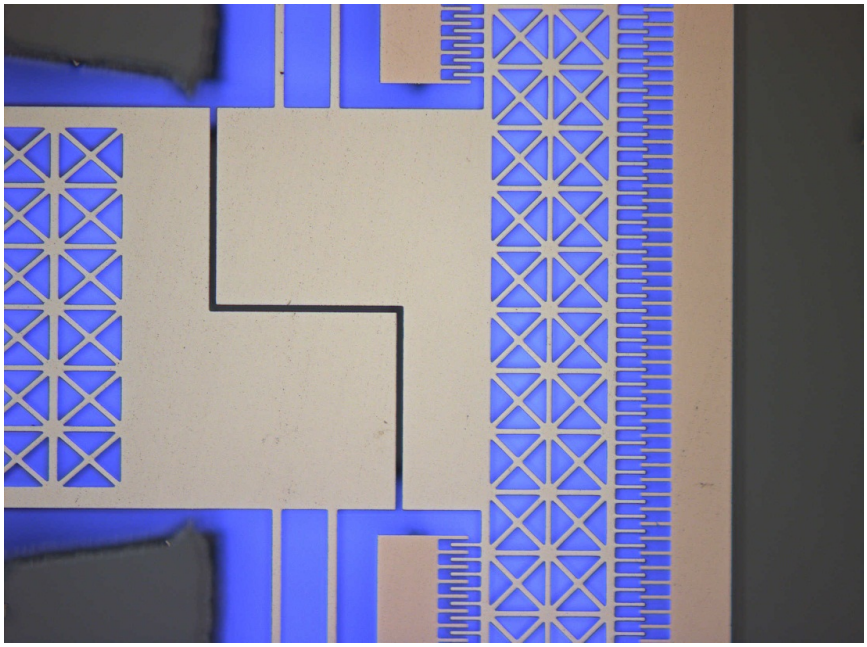

Figure 1.1: Comb drive with electrically separated actuation and sensing combs

existing literatures, the actuating and the sensing combs are electrically connected together. This impose an impediment in circuit design since it requires high actuation voltage (typically around 50V) while the sensing signal usually is less than 5V. In the device demonstrated in this article, the actuating and the sensing combs are separated electrically by having two layers. The conductive layer of both combs on the top are disconnected with the insulator (SiO_2) layer in the bottom (see Figure 1.1). By doing this, the actuation and the sensing combs are decoupled electrically though connected physically. In addition, a pair of differential sensing combs has also been fabricated, which enables a differential sensing scheme and thus enhances Signal-to-Noise-Ratio (SNR) which results in obtaining more accurate model of the system. In this work, a universal capacitive readout IC with a high bandwidth(up to 8kHz), differential input, and high resolution($4aF/\sqrt{Hz}$) is implemented as a sensor.

Proportional-Integral-Derivative (PID) controller, since its first introduction in 1939, has been widely used in both industry and academia. A number of advanced controllers have also been developed and implemented, such as adaptive control [7] and iterative learning con-

trol(ILC) [8]. However, PID controllers are still dominant in industry and most of academia outside of control community for its simplicity in both design and implementation. Recently, H_∞ control have been reportedly implemented successfully in many areas, such as parallel kinematics nano-positioner [9]. Compared to other advanced control techniques, H_∞ control is as simple as PID to use while providing a systematic way of designing. For example, one only needs to assign the desired sensitivity function S, complementary sensitivity function T, and reference-to-controller-output transfer function KS. These transfer functions are defined by

$$\begin{aligned} S &= \frac{1}{1+GK}, \\ T &= \frac{GK}{1+GK}, \\ KS &= \frac{K}{1+GK}, \end{aligned} \tag{1.1}$$

where G, K indicate the plant to be controlled and the controller respectively under the unite feedback configuration (See Figure 1.2).

By convex optimization, a controller that achieves approximate desired performance is readily obtained (see the standard configuration in Figure 1.3). In this work, H_∞ controller is implemented and produces better results compared to PID controller without increasing complexity of controller design or implementation. The H_∞ control algorithm is designed in Matlab and implemented using dSpace processing board. The bandwidth of the closed-loop system via sensitivity function S, namely ω_S, around 67 Hz is achieved with a high robustness to model uncertainty. The bandwidth via complementary sensitivity function T, ω_T, is around 74 Hz, which indicates good noise attenuation by the close loop system.

This article is organized as follows: In Chapter 2, mathematical models for electrostatic comb drives and universal capacitive readout IC are derived, the hardware setup for voltage amplification and embed controller are described, the model of the device to be control

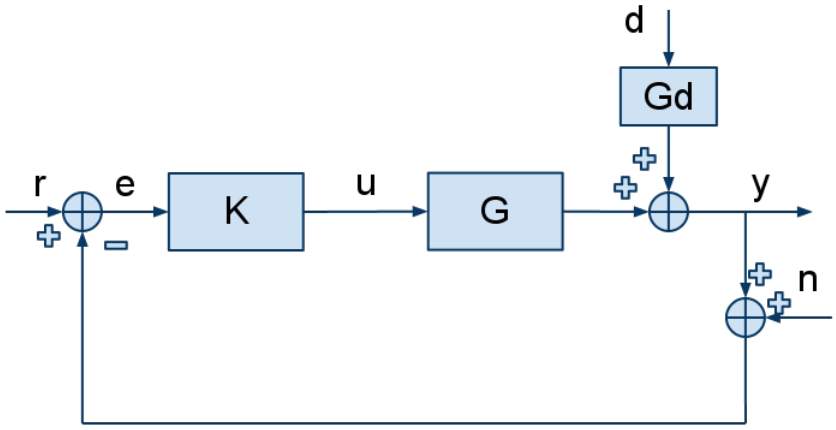

Figure 1.2: Unit feedback configuration: r is the reference signal, e is the tracking error, u is the control effort, d is disturbance, y is output signal, n is measurement noise

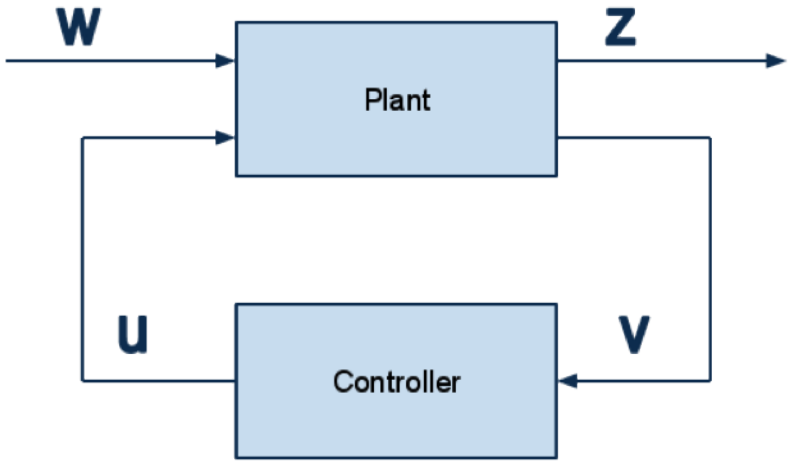

Figure 1.3: Standard configuration of H_∞ controls, w is the exogenous input into plant including reference and disturbance, u is the manipulated input, z is the error signals to be minimized and v is the measured output fed into controller

is parametrically identified based on frequency-domain system identification method and then this identified model is further simplified by a model reduction method. In Chapter 3, system setup is re-visited. Both PID and H_∞ control design processes are discussed and compared. The controller implementation and experimental results are also included along with the corresponding analysis. In Chapter 4, conclusions are drawn from results shown in Chapter 2 and Chapter 3. Future directions are also discussed.

Chapter 2

Model Identification

2.1 Device Description

The capacitive structure to be controlled is presented in Figure 2.1. It contains two linear electrostatic comb drives. Each of them provides motions to the end effector, named XY stage, which has an L shape as shown in Figure 2.1. The motions in the two directions, X and Y, are decoupled from effect of each other, due to the hinges that link the comb drives to the stage. Applied actuation voltage for comb drives can be up to 100 volts. Force generated from electrostatic effects between interdigitated combs drives the stage back and forth. On the other end, sensing combs, which are similarly fabricated as the actuating combs, detect the motion of the stage and transduce this motion into a linear change in capacitance. The capacitance change is further detected by capacitive sensor MS3110.

MS3110 Universal Capacitive Readout IC[1] is a commercial-off-the-shelf capacitance-to-voltage chip, with programmable bandwidth from 0.5 kHz to 8 kHz and resolution $4aF/\sqrt{Hz}$. An important feature of this chip is that it provides differential mode which allows users to directly collect the feedback signal that is proportional to the difference of the two inputs. With this mode combined with the special fabrication which renders a pair of differential sensing combs, the measurement sensitivity is significantly improved. However, the measured capacitance of a single set of sensing comb is the sum of the capacitance of the sensing comb itself, that of the parasitic capacitor and so on. However, this is not a big issue as long as the sensor reading is linear with capacitance change which is further linear with

[1]Commercial Trademark of Irvine Sensors Corporation

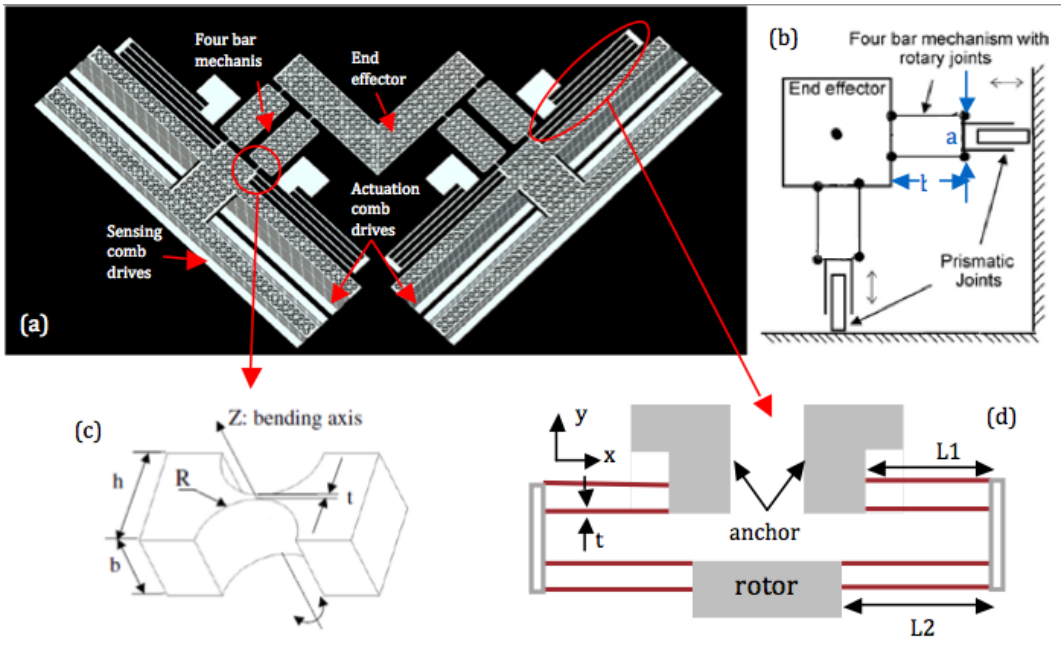

Figure 2.1: SEM pictures for micropositioning XY stage. (a) Overall structure; (b) Actuation/Sensing comb drives; (c) Hinges (d) Actuation comb drive; (e) Differential comb drives for sensing.

comb movement along each axis. The capacitive change rate, $\Delta C/\Delta X$, is $0.0136 pF/\mu m$ [10]. The sensor itself has a resolution of 6.58 nm over a bandwidth of 500 Hz.

2.2 System Setup

The feedback signal is fed back into dSpace(DS1104) controller[2] with 16 bit A/D conversion modules incorporated. The DS1104 also embeds with real-time interface that communicates with computer and D/A conversion modules. In addition, a power amplifier takes the control command from controller and supplies driving voltage to comb drive of the device. As shown in Figure 2.2, a battery is used for powering MS3110 in order to minimize line noise coming through the sensor itself. The two ports in the sensor, CS1 and CS2, are used for the differential mode. CS2 is linked to incremental sensing comb and CS1 is linked to decremental sensing comb. The sensor reading is thus proportional to the difference

[2]Commercial Trademark of dSPACE Inc.

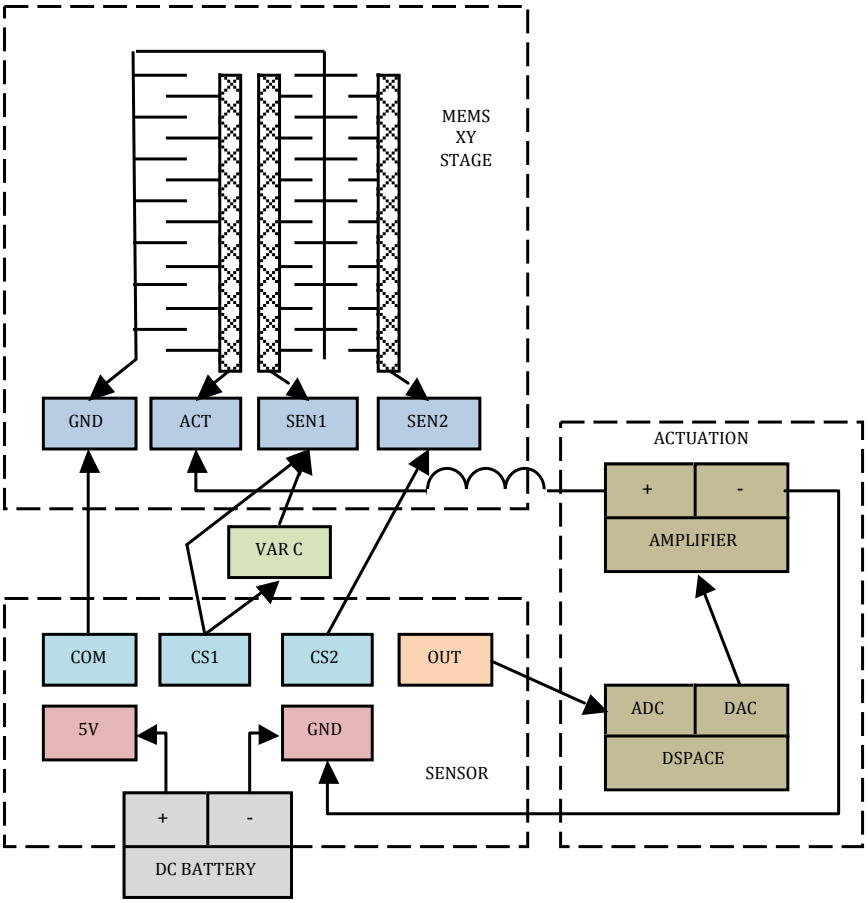

Figure 2.2: Circuit diagram for XY stage test

of capacitance attached to CS2 and CS1. An offset capacitor with fixed capacitance is connected to CS1 in parallel with the decremental comb, in order to offset the difference between each port into the range of sensor measurement (less than 10pF).

The dSpace offers Real-Time Interface and drivers that allow user to design control system in Simulink[3] and then compiles it into C code. The code is then downloaded into and run by the dSpace embed controller. Furthermore, dSpace is linked with ControlDesk[4] which is a software based on the idea of visual instrument and provides a universal experiment

[3]Commercial Trademark of Mathworks Inc.
[4]Commercial Trademark of dSpace Inc.

environment in which users can adjust parameters built into controllers on a real-time basis.

2.3 Mathematical Model

The force that drives combs and thus MEMS stage is given by

$$F = n\frac{\epsilon t}{g}V^2, \tag{2.1}$$

where n is the number of fingers in the comb drive. ϵ is the permittivity of free space, t is the thickness of interdigitated fingers, g is the gap between fingers and V is supply voltage. The dynamics of the electrostatic drive is governed by a second order *Mass-Spring-Damping* system [3] as follows

$$f = m\ddot{d} + b\dot{d} + kd, \tag{2.2}$$

where f is the input force, m is the mass, b is the damping coefficient, k is the spring constant, and d is the length that comb has travelled. By applying Laplace transform the dynamic equation in (2.2) becomes

$$\frac{D}{F} = \frac{1}{ms^2 + bs + k}, \tag{2.3}$$

Alternatively, since the electrostatic driven MEMS stage is made to be an *underdamped* system, the transfer function can be written as

$$\frac{D}{F} = \frac{K\omega_n^2}{s^2 + 2\epsilon\omega_n s + \omega_n^2}, \tag{2.4}$$

where K is the DC gain, ϵ is the damping ration, ω_n is the natural frequency of the MEMS stage. Further the linear translation of comb drive results in capacitance change of the

sensing comb, which follows

$$\Delta C = n\frac{\epsilon t}{g}d, \tag{2.5}$$

where n is the number of fingers in the sensing comb, ϵ is the permittivity of free space, t is the thickness of interdigitated fingers, g is the gap between fingers. The sensor output is linear with the input capacitance [11]. The output voltage of sensor with respect to input capacitance is given as

$$V = G \cdot \frac{CS2 - CS1}{C_f} + V_{off}, \tag{2.6}$$

where V is the sensor output, $CS1$ and $CS2$ are the differential input capacitances, C_f is the feedback capacitance inside sensor which is programmable, G is the programmable gain, V_{off} is programmable offset voltage.

Therefore in this setup, the driving force generated by the driving comb is proportional to V^2. The mechanical response of XY stage is essentially an underdamped second-order system. The linear translation of the comb result in a linear change of capacitance, which is further measured by capacitive sensor MS3110.

2.4 Model Identification

With the theoretical model developed in Section 2.3, we could see that dynamic system with input V^2 and output as the displacement of electrostatic comb is ideally not only a linear system but also an underdamped second order one. It is relatively easy to build up a closed loop system around a plant which is an underdamped second order system. However, other factors, such as fringe effects and parameter inaccuracy within the model, require an experimental way to parametrically identify the model. Although nonlinearities might exist inside the device, for example, the fringe effects and discretization of analog signals, the linear system theory tells us that as long as the perturbation around the designated operation point is small enough, even a nonlinear system could be treated as a linear one.

There are several ways of system identification for linear systems, both in time domain and frequency domain.

2.4.1 Time Domain System Identification

Linear time invariant systems can be represented by state space. The dynamics of linear systems could be described by

$$\dot{x} = Ax + Bu,\ y = Cx + Du, \tag{2.7}$$

where x is the vector of states inside the system, u is the vector of input effort, y is the observed system output. A, B, C, D are constant matrices.

In order to identify the dynamical system, it is necessary to estimate A, B, C, D experimentally. The only source of information is from the system output measured through experiment. Let Δt denote the sample time of digital system used for system identification. By analyzing the corresponding system output $y(0)$, $y(\Delta t)$,...,$y(N\Delta t)$ resulting from a given sequence of input effort $u(0)$, $u(\Delta t)$,...,$u(N\Delta t)$, quite a few method labelled as *Time Domain System Identification* have been discovered, such as step response analysis and Pseudorandom Binary Sequence(PRBS) [12]. The common philosophy behind the methods is already stated above, which is analyzing system output corresponding to specially designed input effort. For a simple example, consider identifying a first-order system which has state-space equations given by

$$\dot{x} = -ax + bu,\ y = cx, \tag{2.8}$$

where a, b, c are positive scalars. By step response analysis shown in Figure 2.3, we could obtain time constant τ and DC gain K for this first-order system which are sufficient for description of the state space equations.

For more complicated systems, we need a parametric way to identify all the system parameters. For complex systems that are being used today, the discretized version of

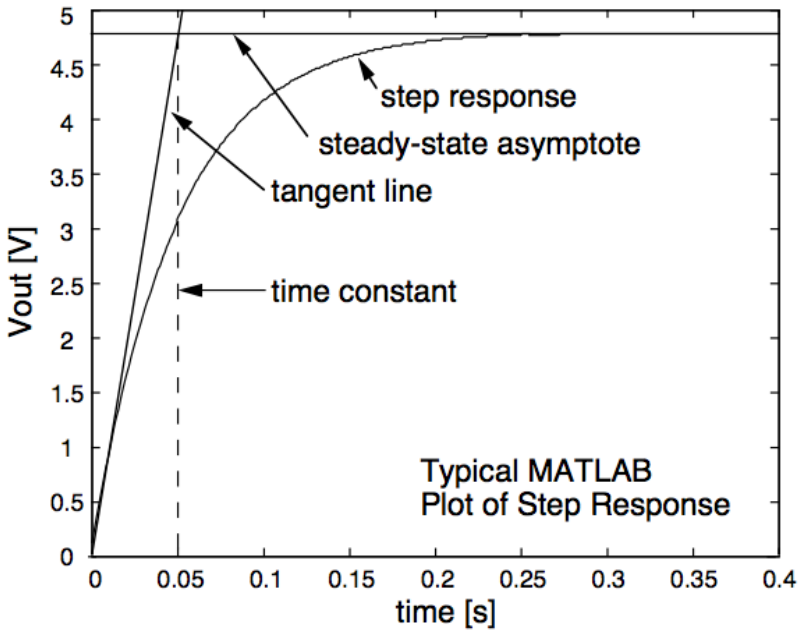

Figure 2.3: Step response-based system identification of 1st order system

governing equations of the form

$$x((n+1)\Delta T) = Ax(n\Delta T) + Bu(n\Delta T),\ y(n\Delta T) = Cx(n\Delta T) + Du(n\Delta T), \tag{2.9}$$

are used instead of the continous-time model. One of these time domain method, namely, Pseudorandom Binary Sequence(PRBS) method utilizes signals randomly shifting between two levels as input signal shown in Figure 2.4, and corresponding system output in order to estimate system parameters such as A, B, C, D. Refer to Chapter 5 in [12] for more details about PRBS.

2.4.2 Frequency Domain System Identification

For time-invariant systems, besides state space equations, we could also use the transfer function to represent the dynamic system. The *Laplace Transform* that transforms time

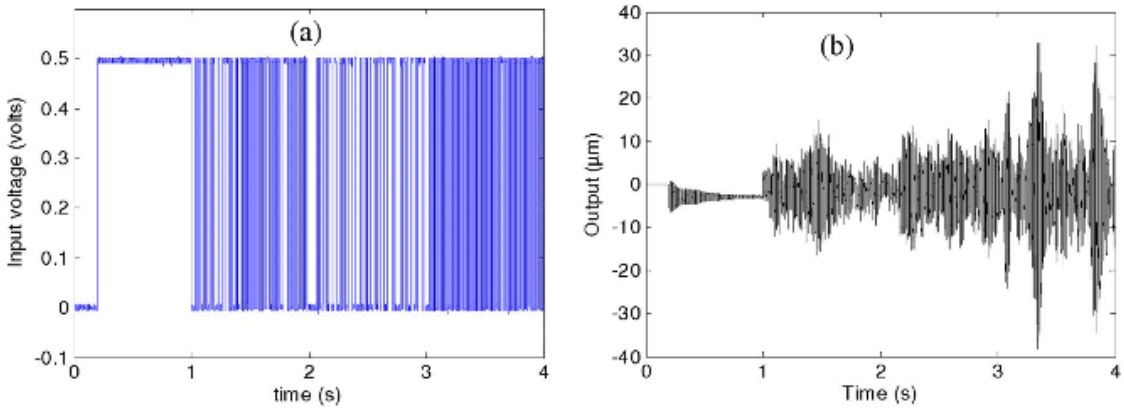

Figure 2.4: a) Pseudorandom Binary Sequence as input signal; b) Corresponding output.

domain function $f(t)$ into complex plane domain function $F(s)$ is given by

$$F(s) = L[f(t)] = \int_0^{+\infty} e^{-st} f(t)\, dt, \tag{2.10}$$

where s is a complex number. By doing this, we obtain transfer function $H(s)$ from dynamic equation governing a linear system. $H(s)$ is given by

$$H(s) = \frac{Y(s)}{U(s)}, \tag{2.11}$$

where Y is system output and U is system input. For MIMO (Multi-Input-Multi-Output) system, the transfer function is a matrix function of s rather that a scalar one for SISO (Single-Input Single-Output) system.

Transfer function is very useful in control engineering, since it provides intuitive indications for system dynamics, such as stability and spectral characteristics. Indeed, one could simply replace s in transfer function with imaginary number $j\omega$ and obtain the spectrum of the dynamical system. The magnitude and phase shift are obtained from the transfer

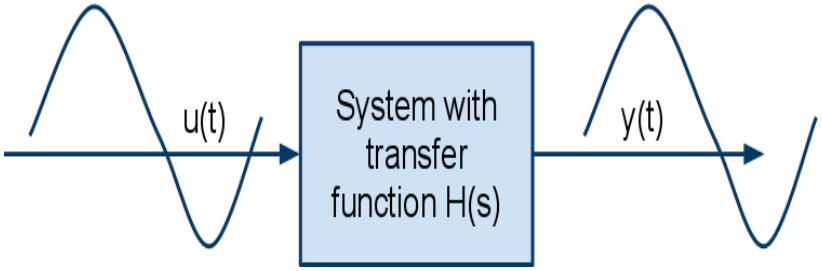

Figure 2.5: Sine Sweep Method for system identification: $u(t)$=$sin(\omega t)$, $y(t)$=$|H(j\omega)|sin(\omega t + 2\pi \angle H(j\omega))$.

function as shown by

$$Magnitude = |H(j\omega)|,\ Phase = \angle H(j\omega). \tag{2.12}$$

Frequency Domain System identification is thus used to obtain system response (Magnitude and Phase shift) at selected frequencies and then estimate the transfer function of linear systems. There are several methods based on this philosophy, such as White Noise Method and Sine Sweep Method. The White Noise Method is inspired by the fact that the spectrum of the white noise contains equal energy within each frequency band. Thus the spectrum of corresponding system output has the same shape as that of the system. On the other hand the Sine Sweep Method is to generate sinusoidal signals and obtain system responses at specific frequencies as shown in Figure 2.5

The latter method is adopted by Dynamic Signal Analyzer(DSA) as the main means for system identification. In this work, system identification of the model relies on DSA from

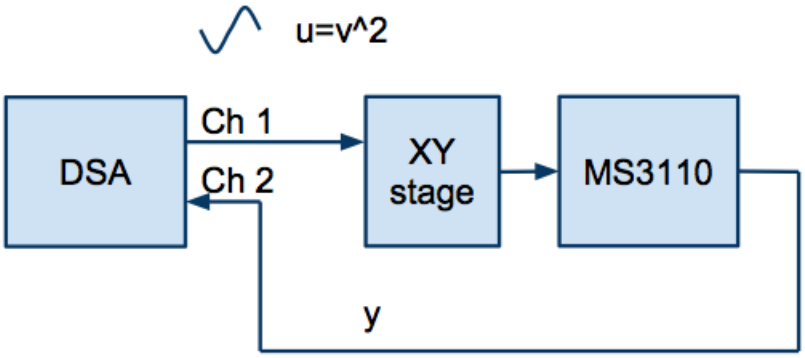

Figure 2.6: Experiment Setup for System Identification

HP Inc. and thus identified model is obtained by Sine Sweep Method.

2.4.3 Experiment Based on Sine Sweep Method

As shown in Figure 2.6, Channel 1 of Dynamic Signal Analyzer (DSA) sends out sinusoidal signals with frequencies ranging from, for example, 1 Hz to 1 kHz. This signal is serving as V^2 where V is the supply voltage to comb drive. In order to account for this, we can not simply feed signal from channel 1 directly to comb drive but instead have to modify it in dSpace. Within dSpace, first take the square root of the signal from Channel 1, let's say v_1, and then multiply $\sqrt{v_1}$ by 1/200 in order to neutralize the scaling factors from D/A conversion and power amplification. Note that there is also a scaling factor of 1/10 during A/D conversion. Therefore in dSpace programing, input from A/D conversion is multiplied by 10 in order to recover the signal v_1 from channel 1.

The supply voltage sent out to the comb drive is essentially $\sqrt{v_1}$. As shown in Equation (2.1), the system is a linear one with input as V^2, which is essentially v_1, and output as the sensor reading which is sent back to DSA through channel 2. DSA estimates the system response (magnitude and phase shift) at each chosen frequency. For example, we could simply program DSA to send out sinusoidal signals with frequencies sweeping from 1

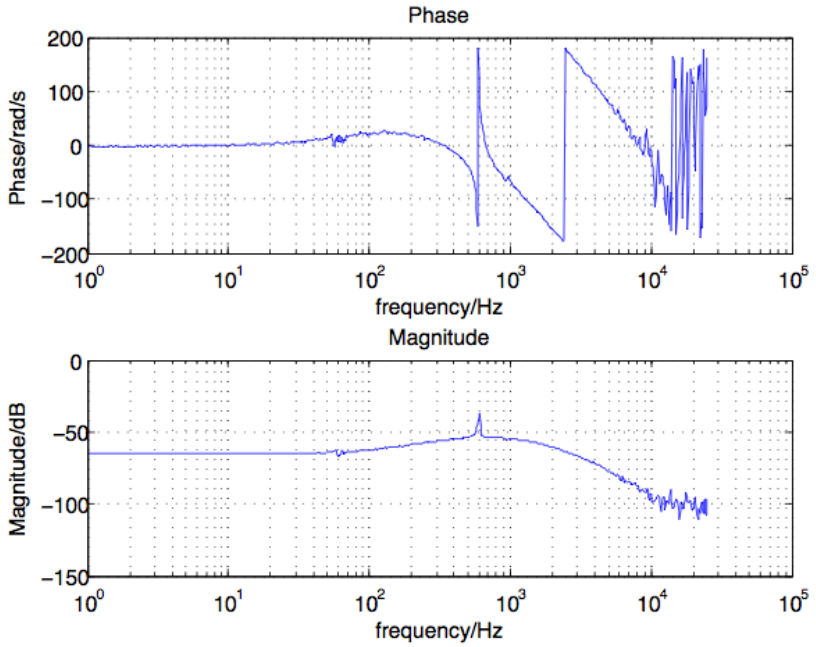

Figure 2.7: Bode plot from sine sweep

Hz to 1 kHz with resolution of 200 points per sweep, equally distributed by either logarithm scale or normal scale.

The data rate of dSpace can be up to 50 kHz. Therefore in the experiment, the sweeping range is set to be 1 Hz to 25 kHz in order to prevent aliasing. The frequency resolution is set to be 100 points per decade. The result is shown in Figure 2.7.

Based on the mathematical model previously derived, the system (XY stage+sensor) ideally is an underdamped second order system with supposed-to-be low damping ratio. Based on the experiment results shown in Figure 2.7, the peak in magnitude appears at the frequency of around 600 Hz and the magnitude(around 10dB) of the peak indicates a small damping ratio. The phase continues to drop down as the frequency increases while the mathematical model predicts that the phase eventually converges monotonically to $-180°$. Indeed this is understandable since there is always time delay in digital system such as dSpace. Suppose the time delay is τ, as the frequency f of sinusoidal signals increases, the corresponding phase delay by dSpace $\tau \cdot f \cdot 360°$ also increase creases linearly with frequency

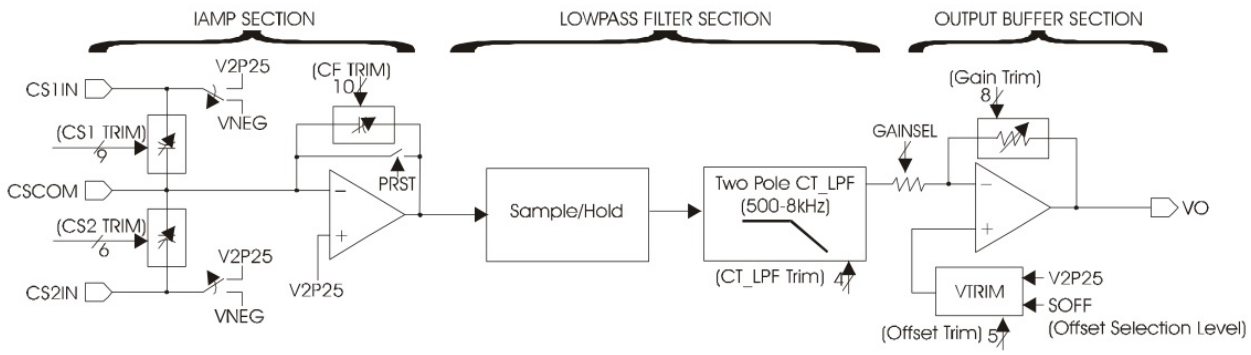

Figure 2.8: Block diagram of MS3110

f (in logarithm scale, exponentially), which can be verified in the Figure 2.7. Another issue is the zero appearing around 80 Hz as shown in Figure 2.7. This is not expected from XY stage since ideally it should be a second order system without zeros.

However if we take a look at the internal structure of the capacitive sensor, the square waves (marked as V2P25/VNEG in Figure 2.8) sent through external capacitors has a frequency around 100 kHz and the sensor output is proportional to the change rate of charge stored in external capacitor, for example, CS1. This part of dynamics is governed by

$$V \propto \dot{q} = c\dot{u} = c\dot{u} + \dot{c}u, \tag{2.13}$$

where c is the capacitance to be measured and u is the voltage of the square wave.

Therefore in the transfer function of sensor, a zero is brought in due to the additional item $\dot{c}u$ in Equation (2.13). Therefore, in the closed loop design, the operating region of the closed loop system does not exceed 80 Hz since beyond that the sensor reading itself is not accurate in terms of measuring capacitance due to limitation on the dynamics.

2.4.4 Model Fitting

The result from experiment only contains informations of the system responses at chosen frequencies with the parametric model yet to be identified from experiment data. Partic-

ularly, a transfer function is needed to fit into data of frequency responses obtained from DSA.

From bode plot shown in Figure 2.7, we make an estimation of the model as follows: first, a low pass filter from sensor consisting of one zero and two real poles is expected as we could see in Figure 2.7; The XY stage contributes a pair of complex poles. Thus ideally, the model should have 1 zero, 2 real poles, and a pair of complex poles. However, from the bode plot obtained from experiment, the number of poles exceeds that of zeros only by 2 since the magnitude of transfer function at high frequency decaying at slope of -40 dB/dec. In addition, the shape of the peak in magnitude and corresponding phase delay suggest that there exists higher order dynamics at high frequencies, essentially beyond $80Hz$.

For system of order higher than 2, usually it requires computer algorithms to estimate the parametric model. Basically what we got from this experiment are complex frequency responses at a finite set of chosen frequencies. What we need to do is to fit a transfer function, or equivalently, a complex polynomial curve to experimental data. Based on algorithms proposed in [13] and [14], a *matlab* function named *invfreqs* is used to identify continuous-time filter parameters from frequency response data, which is essentially fitting a transfer function into experimental frequency response data.

The common used inputs for *invfreqs* are frequency response data, orders of numerator and denominator. The first one, that is frequency response data, has already obtained. Based on the analysis of bode plot in the previous paragraph, the order of denominator in the transfer function exceeds that of numerator by 2. As a result, a transfer function with 8th order numerator and 10th order denominator is fitted into frequency response data up to 5 kHz, since beyond 5 kHz noise and lack of sample points per period flaw the measurement. The returned transfer function fits well with frequency response data as shown in Figure 2.9. Model is also automatically stabilized by algorithms inside *invfreqs*.

The fitting transfer function is given by

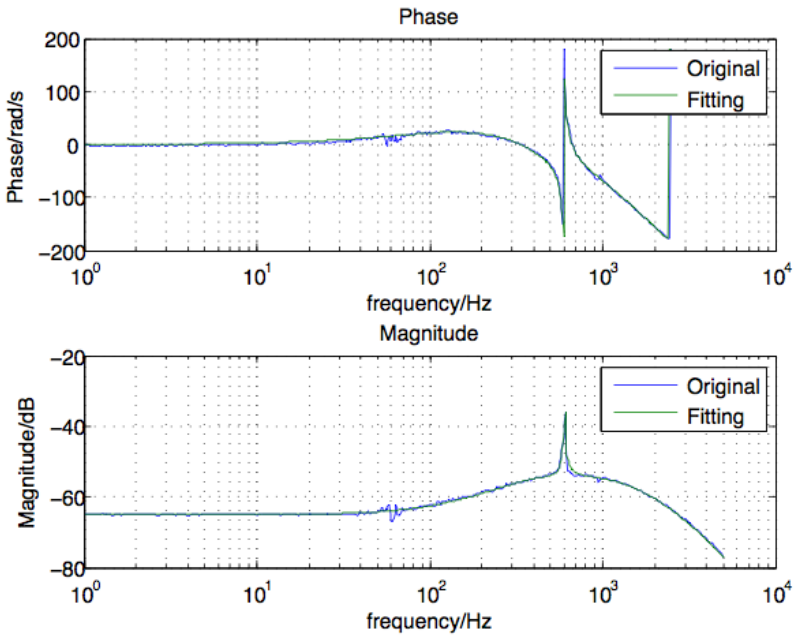

Figure 2.9: Transfer function fitting

$$
\begin{aligned}
A &= 1.087e004s^8 - 2.963e009s^7 + 1.31e014s^6 + 6.538e017s^5 + 3.888e021s^4 \\
&\quad +2.157e025s^3 + 3.625e028s^2 + 1.664e032s + 1.1e035, \\
B &= s^{10} + 5.729e004s^9 + 1.863e009s^8 + 2.656e013s^7 + 2.448e017s^6 + 1.364e021s^5 \\
&\quad +6.914e024s^4 + 2.339e028s^3 + 6.788e031s^2 + 1.309e035s + 1.95e038, \qquad (2.14)
\end{aligned}
$$

where A is the numerator and B is the denominator. All the poles are stable (see Figure 2.10).

2.4.5 Model Reduction

The model identified is of 10th order, which adds complexity to designing corresponding controller based upon H_∞ method. For example, if the plant is of 10th order, the corresponding H_∞ controller usually is of 10th order or even higher. Thus it is necessary to

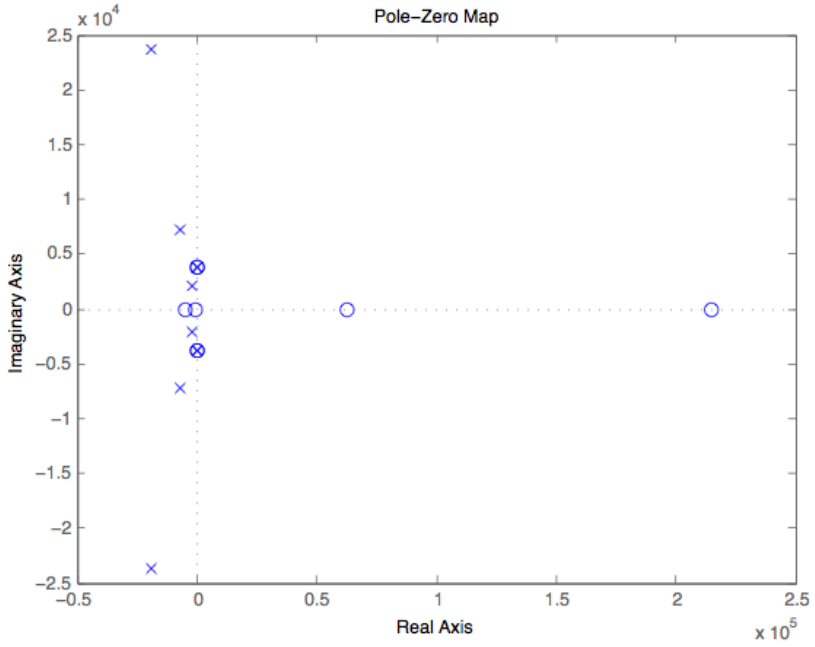

Figure 2.10: Poles(X) and zeros(O) on complex plane

implement *Model Reduction* technique to reduce computational complexity, which is crucial for high-speed controller implementation.

One of these techniques, named *Balanced Realization*, uses *hankel singular values* σ_i of transfer function, say H(s), as a measure of importance for each corresponding state x_i in terms of contribution to input-output behavior [15]. By implementing balanced realization, the importance of each state could be easily estimated and then it is possible to remove certain states from the original model without significant change system behavior.

By calling matlab function *balreal*, a vector consisting of hankel singular values is obtained and shown in Figure 2.11. As it shows, the last two singular values, which belong to 9th and 10th state respectively, are negligible compared to other singular values. This means that we could simply remove the last two states in the original model and still maintain almost the same system response.

By calling matlab function *modred*, the last two states are removed and the corresponding system response is shown in Figure 2.12 along with those from the experiment and fitting

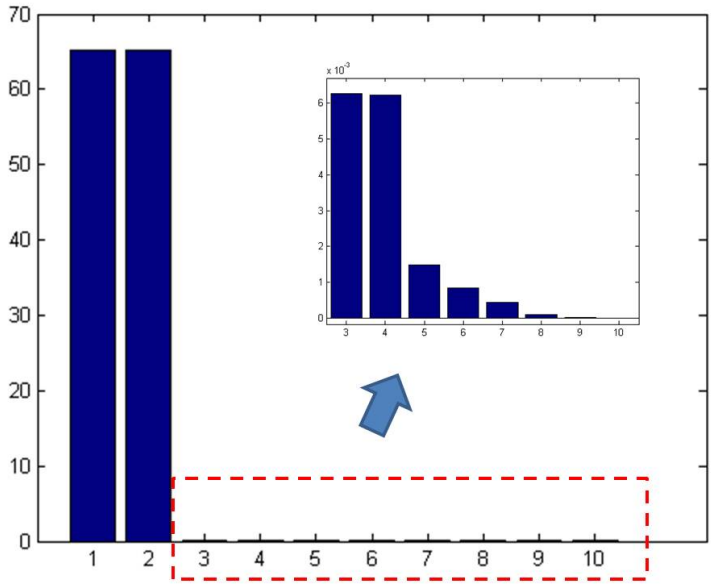

Figure 2.11: Hankel Singular Values of H(s)

model. As we can see here, the reduced model matches well with both experimental system response and fitting model, which is prediced based on results from balance realization.

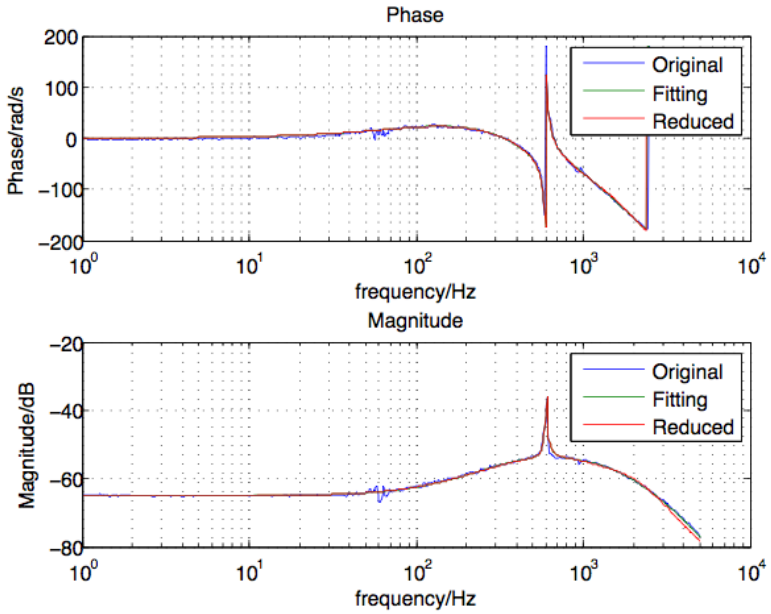

Figure 2.12: Bode plots of transfer functions from original experiment, fitting model, and reduced model

Chapter 3

Control Design and Implementation

Once system identification is done, what remains is control design and implementation. As shown in Figure 3.1, square root, power amplifier, XY stage, and sensor blocks constitute the model identified in Chapter 2. Physically, the controller, square root, reference signal generator and subtraction blocks all reside by dSpace process board.

Figure 3.1 shows the closed loop configuration. The next step is the design of a controller that achieves good closed loop bandwidth, stability, resolution, and robustness. Over the past half century, a lot of control theories have been developed to address these issues inside control systems. For example, Proportional-Integral-Derivative, also known as PID, control is widely used in industry since its introduction in 1939. Other more advanced control theories, such as adaptive control, achieve very high control performance since controllers are being adaptively modified based on feedback information [7]. Other control theories, for example, iterative learning control(ILC), design controllers in such a way that it runs the

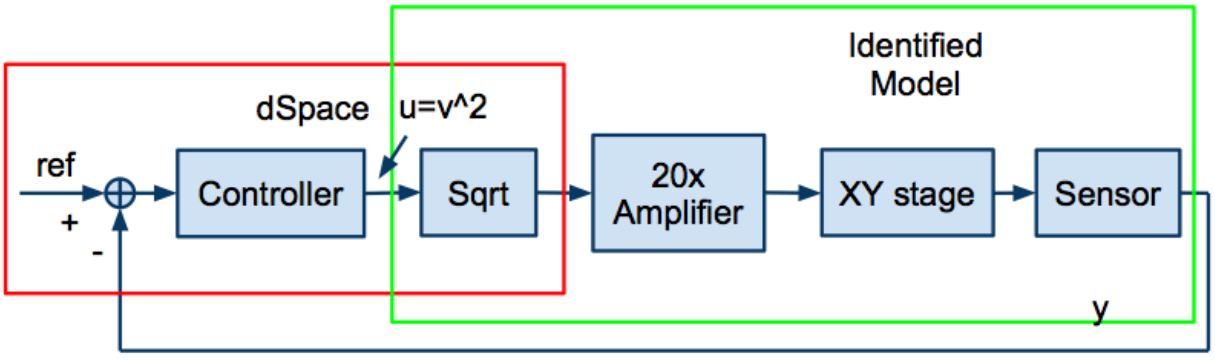

Figure 3.1: Block Diagram of closed loop System

whole process for multiple times and the controller learns from the past performance [8]. ILC is extremely useful for massive production process since it can afford some failure in production at the very initial stage. For example, it is not an issue to have 100 initial products fail while the remaining 9900 are precisely manufactured. Usually, advanced control methods renders better performance than PID but are much more complicated to learn and implement. Therefore, PID is still dominant in industry and other more advanced control theories are yet to gain more attention from control applications. Recently, H_∞ control have been reportedly implemented successfully in many areas, such as parallel kinematics nano-positioner [9]. Compared to other advanced control techniques, H_∞ control is as simple as PID to use while providing a systematic way of designing. For example, one only need to assign the desired sensitivity transfer function S, complement transfer function T, and reference-to-controller output transfer function KS. By using standard algorithms that solve convex optimization, a controller that achieves approximately the desired performance is readily obtained. In this chapter, both PID and H_∞ control have been designed and implemented for controlling MEMS probing device. By doing this, H_∞ control is compared with traditional PID control and then the advantages of H_∞ control are demonstrated.

3.1 PID Control

3.1.1 Introduction

The PID controller is given by

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)\, d\tau + K_D \frac{de}{dt}, \tag{3.1}$$

where $u(t)$ is the control effort at time t, $e(t)$ is the error(reference signal minus feedback signal). K_P, K_I, and K_D are gains for proportional, integral, and derivative actions.

Equivalently, PID controller could be expressed in terms of transfer function as follows,

$$\frac{U(s)}{E(s)} = K_P + K_I\frac{1}{s} + K_D s. \tag{3.2}$$

The proportional action is simply proportional to error $e(t)$ by a factor of K_P, which is the most intuitive idea in feedback control: the larger the error is, the more effort the controller puts in order to pull feedback signal back to the set point. The integral action, on the other hand, is proportional to the integral of error over time t by a factor of K_I. Essentially integral action is designed to eliminate static error $e(\infty)$ since as long as error is not zero the integral action will be increasing or decreasing control effort accordingly in order to drive feedback signal to the set point. Usually, Proportional-Integral(PI) controller is enough for controlling first order plant with a commonly used version given by

$$u(t) = K[e(t) + \frac{1}{T_i}\int_0^t e(\tau)\,d\tau], \tag{3.3}$$

where T_i is the integral time. As long as T_i is finite, error will eventually decays to zero. The derivative action aims to increase stability margin of feedback systems. Combined with proportional action, the Proportional-Derivative(PD) controller is essentially extrapolating the error by first order approximation. The control effort of PD controllers is determined by

$$u(t) = K[e(t) + T_d\frac{de(t)}{dt}], \tag{3.4}$$

where T_d is the derivative time. By *Taylor Expansion*, the first order approximation is achieved by

$$e(t + T_d) \approx e(t) + T_d\frac{de(t)}{dt}. \tag{3.5}$$

A commonly used version of PID controller is expressed by

$$u(t) = K[e(t) + \frac{1}{T_i}\int_0^t e(\tau)\,d\tau + T_d\frac{de}{dt}]. \tag{3.6}$$

For more details about PID control, refer to [16].

3.1.2 PID Controller Design

There is a vast literature on PID controller design. One of the most notable tuning scheme is the *Ziegler-Nichols* method. It is simple and intuitive for even people with minimal knowledge of control theory to use. Ziegler-Nichols method is a heuristic way to determine parameters for PID controllers. For example, one of Z-N methods is based on frequency response. First, apply proportional only control in the feedback loop. Increase proportional gain K_P to the ultimate value K_u with which the output oscillate with period T_u. Determine corresponding K_P, K_I, K_D based on the Table 3.1

Table 3.1: Controller parameters for the Ziegler-Nichols frequency response method

Controller	K/K_u	T_i/T_u	T_d/T_u
P	0.5		
PI	0.4	0.8	
PID	0.6	0.5	0.125

However Z-N method only gives a good start in the controller design. Empirical tuning is needed to obtain a reasonable closed loop performance. There are several rules of thumb to follow when tuning PID controller according to [16].

- Increasing proportional gain K_P speeds up system response but decreases stability
- Error decays more rapidly if integration time T_i is decreased
- Decreasing integration time T_i decreases stability
- Increasing derivative time T_d improves stability

There are also other advanced methods in PID control designing, such as optimization method, robust loop shaping, etc. according to [16]. We will save this effort for the more

powerful and convenient controller designing method, H_∞ method. In this section about PID control design, empirical tuning method is adopted to design a PID controller since it is simple to use and the system to control is complicated.

3.1.3 PID Controller Implementation

As demonstrated in Figure 2.9, there is a *zero* in the transfer function around 80 Hz, which makes PID controller design a little bit difficult from open loop shaping point of view since we want the open loop transfer function L to cross 0 dB around this frequency region based on the fundamental limitation from sensor discussed in Chapter 2. Instead, a pole around 80 Hz is included in the modified PID controller as follows

$$C(s) = \frac{1}{s/500 + 1} \cdot (K_P + K_I \frac{1}{s} + K_D s), \tag{3.7}$$

where the values of K_P, K_I, K_D are decided by empirical rules as shown in the previous section. First, increase the proportional gain K_P until it almost reaches the ultimate gain K_u. Then increase K_I to eliminate static error until the feedback system almost becomes unstable. Increase K_D to stabilize the system which however proves to be not necessary in this case. Therefore, the PID controller with an impressive performance is presented as

$$C(s) = \frac{1}{s/500 + 1} \cdot (100 + 300000 \frac{1}{s}). \tag{3.8}$$

The resulting feedback system that operates at 10 kHz is thus identified by DSA. Corresponding sensitivity function S and complement sensitivity function T are demonstrated in Figure 3.2 and Figure 3.3 respectively along with their simulation counterparts. As we can see in these two figures, the simulation and experiment matches well up to 500 Hz. Beyond that, the time delay take control of phase delay as predicted.

The closed loop bandwidth defined by S is around 26 Hz while that defined by T is around 32 Hz. Step response and tracking performance are also demonstrated in Figure 3.4

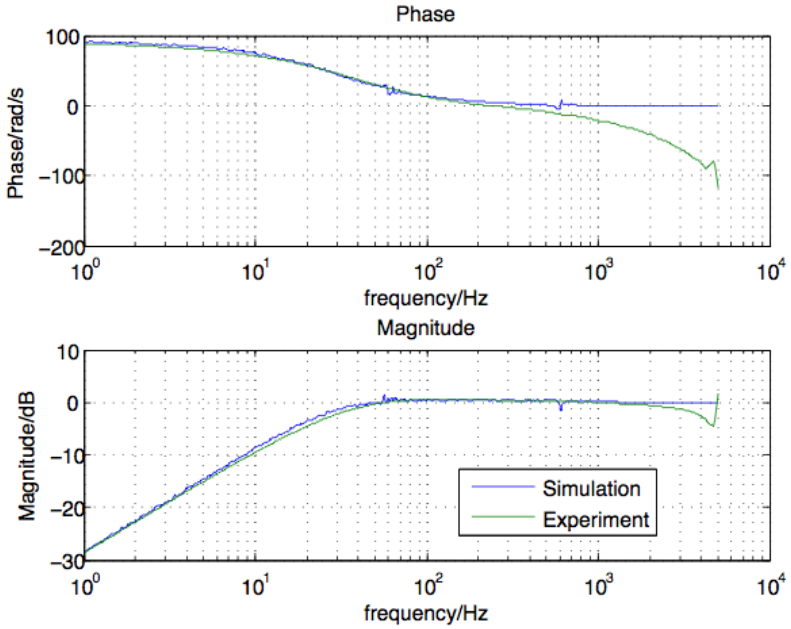

Figure 3.2: Bode plot of sensitivity function S

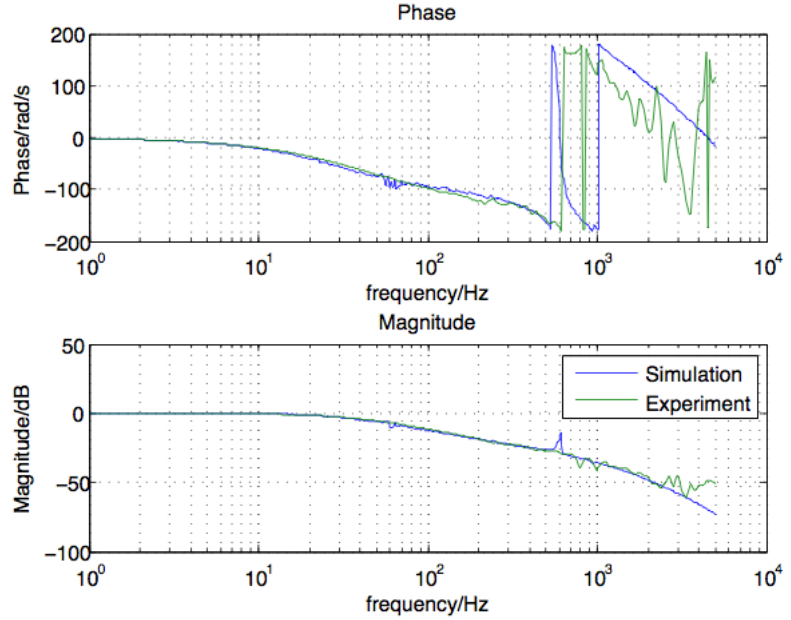

Figure 3.3: Bode plot of complementary sensitivity function T

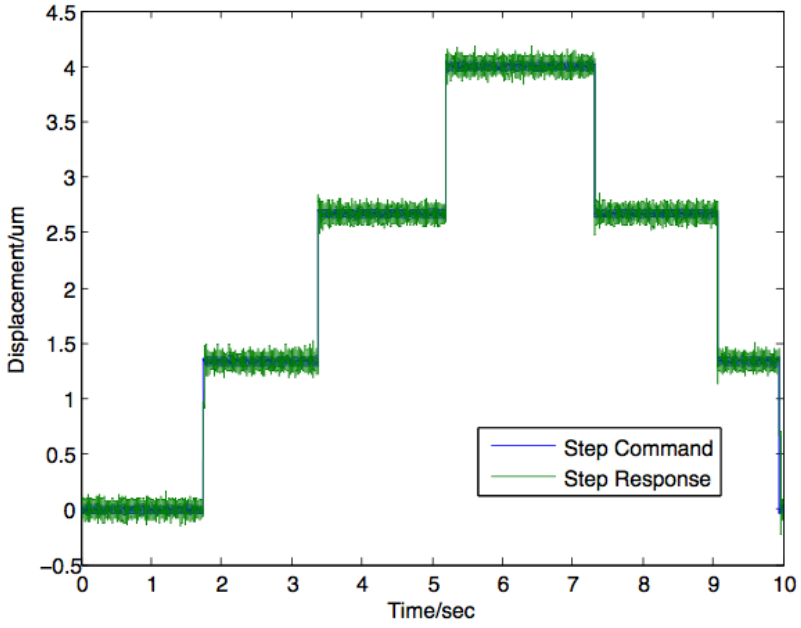

Figure 3.4: Step response of the closed loop system based on PID controller

and Figure 3.5. From results shown here, PID control builds up a feedback system with an impressive closed loop performance. However, there are some disadvantages about PID control. First, usually it requires tedious tuning effort to work out a good controller. Second, the performance is limited by the order of PID controllers. In the case demonstrated here, the PID controller is only consisting of a first order low pass filter and a standard PID controller. With more advanced controller, the closed loop performance is expected to be better. With computerized method introduced in control design such as H_∞ control, more complicated controllers are available for control application to greatly improve control performance while having work load reduced.

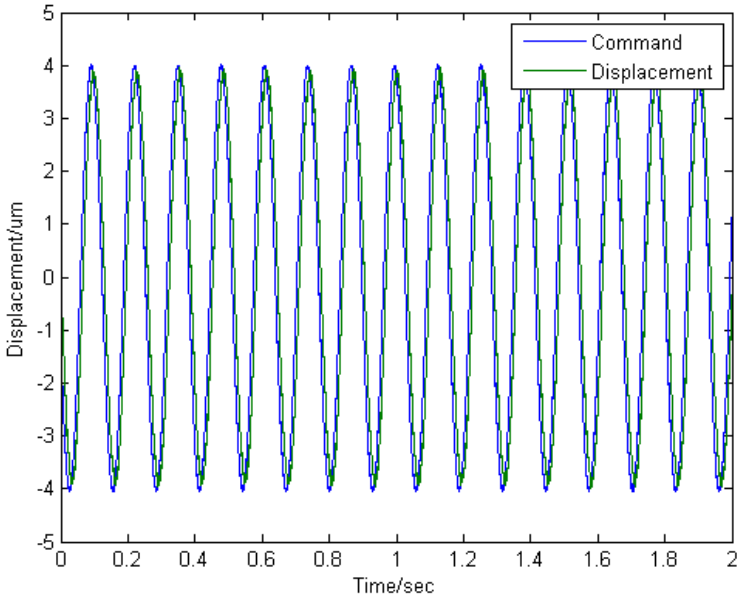

Figure 3.5: Tracking performance of the closed loop system based on PID controller: tracking a 7 Hz sinusoidal command

3.2 H_∞ Control

3.2.1 Introduction

The general setup of feedback system is shown in Figure 1.2. In controller design, the way to estimate its performance usually bases on several transfer functions, such as the sensitivity function S, the complementary sensitivity function T, and controller output transfer function KS. They are defined as follows

$$S = \frac{1}{1+KG}, T = \frac{KG}{1+KG}, KS = \frac{K}{1+KG}, \tag{3.9}$$

where G is transfer function of the plant to be controlled and K denotes that of the corresponding controller.

The effect of reference r, disturbance d, and noise n effect system output y and tracking error e is given by

$$
\begin{aligned}
y &= Gu + G_d d = GK(r - y - n) + G_d d, \\
y &= \frac{GK}{1+GK} r - \frac{GK}{1+GK} n + \frac{G_d}{1+GK} d, \\
y &= Tr - Tn + G_d S d, \text{and} \\
e &= r - y = Sr + Tn - G_d S d.
\end{aligned} \tag{3.10}
$$

Essentially, S is the transfer function from r to e and thus is the primary measure of the tracking performance. The larger the bandwidth ω_s of S is, the better the tracking performance is. Moreover, S also indicates robustness, which is verified by

$$
\frac{dT/T}{dG/G} = S. \tag{3.11}
$$

It indicates that smaller S is, the more robust the feedback system is respect to model uncertainty within G.

Also since

$$
\max_{\omega} |S(j\omega)| = \frac{1}{\min_{\omega} |1 + G(j\omega)K(j\omega)|}, \tag{3.12}
$$

the smaller the maximal magnitude of S is, the more stabilized the feedback system is since that means GK is further away from the critical point *-1* on the complex plane.

The complementary sensitivity function T provides measures for scanning performance and noise sensitivity as T is the transfer function from r to system output y, and also that from noise n to y. Therefore, general objective of control design is to increase the bandwidth ω_s as much as possible while limiting ω_t from exploding up as this introduces more noise. Since we also do not want to put too much control effort to the certain control performance, there we impose a limit on the size of the control effort. As the transfer function KS indicates how reference r determines u, there must be consideration on designing KS too. Usually,

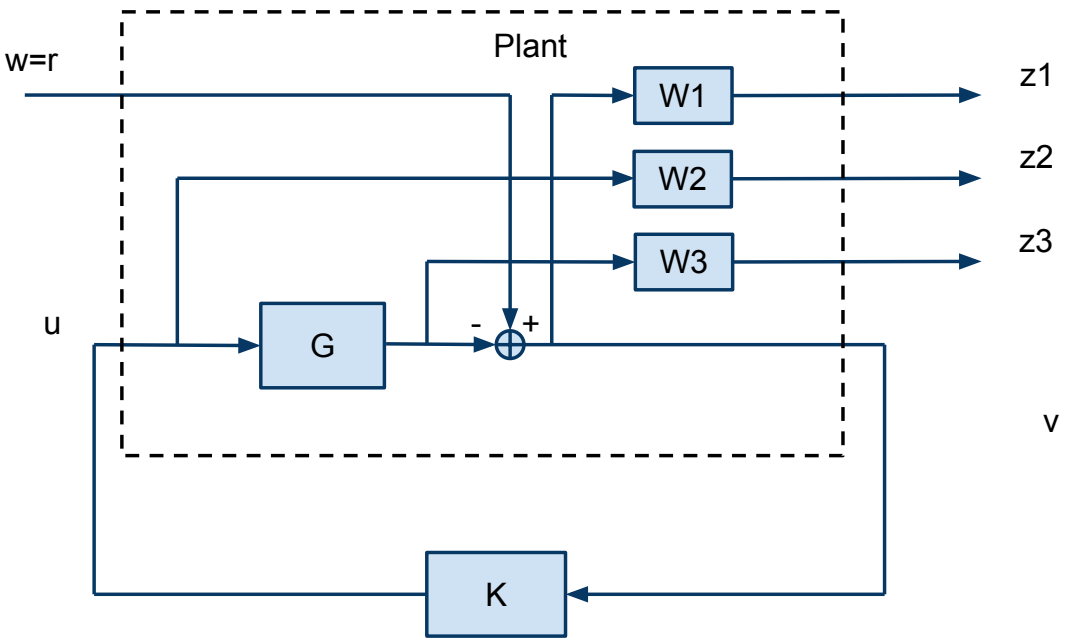

Figure 3.6: $S/KS/T$ mixed-sensitivity minimization in standard form(tracking)

in the design process, there will be an upper bound for KS.

The general setup of H_∞ control design as shown in Figure 1.3 of Chapter 1 could be converted into an optimization problem. One could refers to [17] and [18] for more details about H_∞ control theory.

The weight functions W_1, W_2, and W_3 (see Figure 3.6) are chosen in a way such that the following norm

$$\left\| \begin{bmatrix} W_1 S \\ W_2 K S \\ W_3 T \end{bmatrix} \right\|_\infty , \tag{3.13}$$

is minimized, where $||\circ||_\infty$ is a norm defined as the maximum of the absolute values of all the functions in the vector over the whole domain which, in this case, all frequencies. This means each of $|W_1S|_\infty$, $|W_2KS|_\infty$, and $|W_3T|_\infty$ are bounded by this minimum, named as γ, the minimum over all K. In other words, $|S|$, $|KS|$, and $|T|$ are bounded by $|\gamma/W_1|$, $|\gamma/W_2|$, and $|\gamma/W_3|$ over all frequencies. Therefore, the controller design problem comes down to choosing proper weight functions W_1,W_2, and W_3 and achieving an acceptable γ by available methods that solve the optimization problem over all possible controllers K. These will secure a controller K that renders a desired closed loop system performance in bandwidth, noise reduction , robustness, and stability.

There are commercial softwares dealing with such H_∞ optimization problem available, such as *Matlab*. Indeed, Matlab provides several functions addressing H_∞ control design, such as *loopsyn*, *ncfsyn*, and *mixsyn*. Among them, *mixsyn* provides synthesis of mixed-sensitivities as shown in Figure 3.6 and provides an intuitive way for controller design. Users only need to assign desired weight functions to *mixsyn* which guarantees a desired closed loop performance, and receive a corresponding controller K, and the value γ based on which users estimate how far away the actually sensitivity functions are from desired ones. This process is almost as simple as PID controller design while is very likely to render a better controller.

3.2.2 H_∞ Controller Design

Before starting H_∞ controller design, there are a couple of fundamental limitations on closed loop systems which should be kept in mind.

First of all, S plus T is 1 since $S = (1 + GK)^{-1}$ and $T = GK(1 + GK)^{-1}$. This means there must be a trade-off between S and T since a small S leads to a large T and vice versa.

In addition, in a closed loop system with phase margin (PM) less than 90°, the two bandwidths satisfy that $\omega_S < \omega_T$.

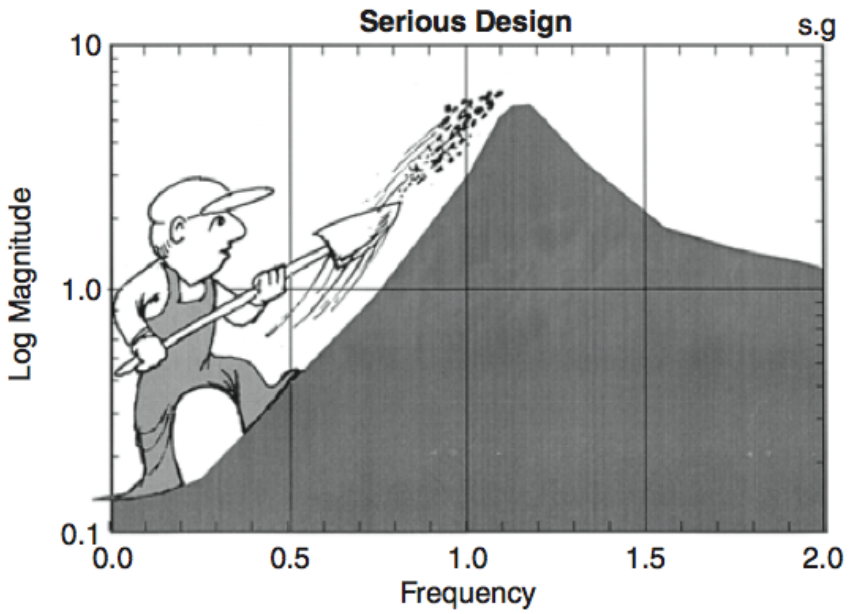

Figure 3.7: Sensitivity reduction at low frequencies unavoidably leads to sensitivity increase at higher frequencies

It has also been proved that the integral of bode sensitivity

$$\int_0^\infty ln|S(j\omega)|\, d\omega, \tag{3.14}$$

must be zero for any stable plant. Therefore it is not possible to make $|S(j\omega)|$ arbitrarily small at low frequencies since this leads to a large *peak* and thus raises issues in stability. This piece of fundamental limitation on control design is well demonstrated in Figure 3.7

There are also other design constraints but the three mentioned above are the very important to follow during design process. Control designs are balanced between bandwidth of S and T, robustness and stability. In control design, there is no single answer but a multiple of them. Thus it is important to cope with preset objectives for closed loop performance. In this case, a PID controller has already been setup. Thus in this work, the design objective of H_∞ control is mainly based on the previous standard and shows significant improvement from PID control.

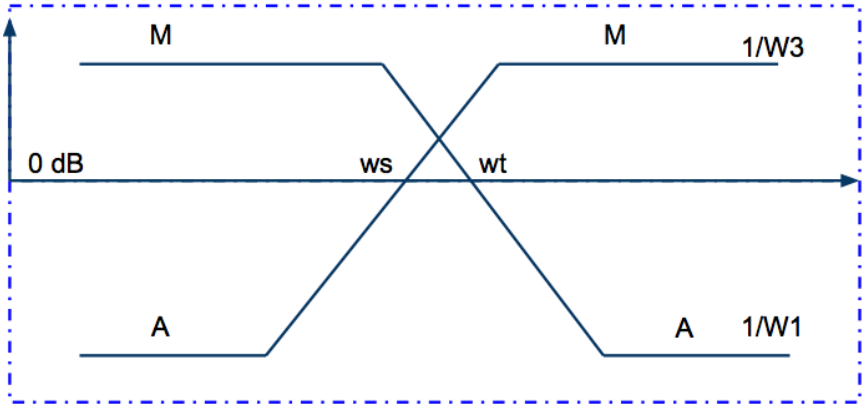

Figure 3.8: Bode plot of weight functions ω_1 and ω_3

The key part of H_∞ control is determining weight functions W_1 and W_3. W_2 is left to be empty in this case since we can always reduce the range of input to cut down control effort if necessary and the disturbance is a very small signal. From the previous discussion, assume γ=1, $1/W_1$ and $1/W_3$ are the desired upper bound for S and T respectively. Therefore W_1 and W_3 are assigned into forms as follows

$$\begin{aligned} W_1 &= \frac{s/M + \omega_S}{s + \omega_S A}, \\ W_3 &= \frac{s + \omega_T/M}{As + \omega_T}. \end{aligned} \tag{3.15}$$

W_1 is equal to $1/A$ at very low frequencies and drops down to $1/M$ at very high frequency. The crossover frequency for W_1 is around ω_S. Similarly, W_3 is equal to $1/M$ at very low frequencies and rises up to $1/A$ at very high frequency. The crossover frequency for W_3 is around ω_T. Once controller mixed-synthesis is finished successfully by convex optimization with $\gamma \approx 1$, corresponding sensitivity function S and complement sensitivity function T are bounded by $1/W_1$ and $1/W_3$ respectively as shown in Figure 3.8. Note that if γ is not 1, then these *upper bounds* should be multiplied by the constant γ.

In this design, choose parameters for desired weight function as follows. Since the zero

from the sensor is around 80 Hz, there is no point of pushing ω_S higher up. Moreover, ω_T as an indicator of noise sensitivity goes up as ω_S increases. Therefore choose ω_S to be around 60 Hz and correspondingly ω_T to be around 70 Hz which leads to good noise attenuation. A small M that is a little bit over 1 means good stability while an infinitely small A leads to good robustness and good noise reduction too.

- A=1/1000
- M=2
- ω_S=60 Hz
- ω_T=70 Hz

Then run matlab command *mixsyn*. A 10th order controller is returned. $\gamma = 1.085937433134438$ means that design objectives are almost accomplished. Simulated closed loop performance is compared with that with PID control in Figure 3.9. Based on this simulation, we expects a better bandwidth as both ω_T and ω_T. From the simulation, ω_S of H_∞ is around 60 Hz compared to that of PID around 26 Hz. ω_T of H_∞ is around 66 Hz compared to that of PID around 32 Hz. Therefore the H_∞ controller has a better closed loop bandwidth ω_S while sacrificing noise attenuation. Closed loop stability is maintained although bandwidth is improved by the new design. $|KS|_\infty$ are still the same for both controllers and thus reference-to-control imposes no further limitation on control implementation. By the new H_∞ control design, a control rendering a better closed loop performance is obtained without increasing work load compared to PID control design.

3.2.3 H_∞ Controller Implementation

The controller is implemented in dSpace process board. Before implementation, the 10th order controller needs to be simplified since dSpace can't run a continuous-time controller of such a high order at a date rate around 10 kHz. Two parts of work have to be done before

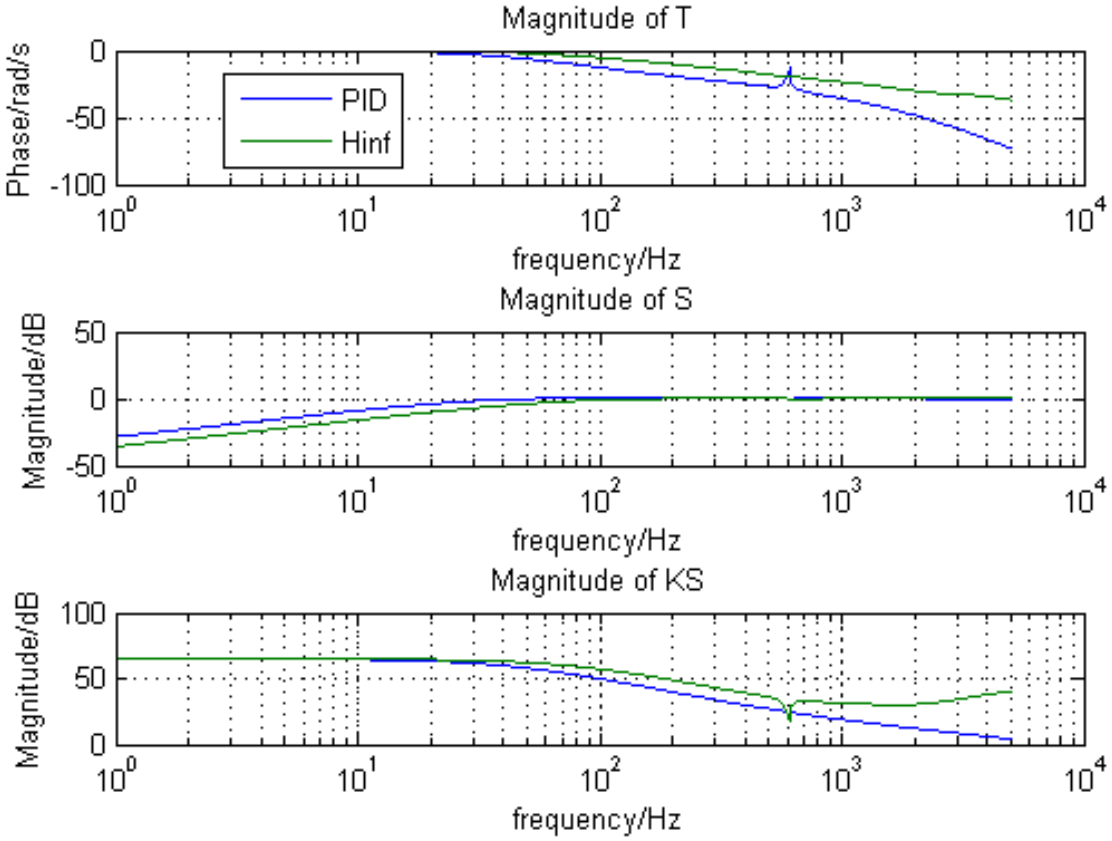

Figure 3.9: Comparison between H_∞ and PID control

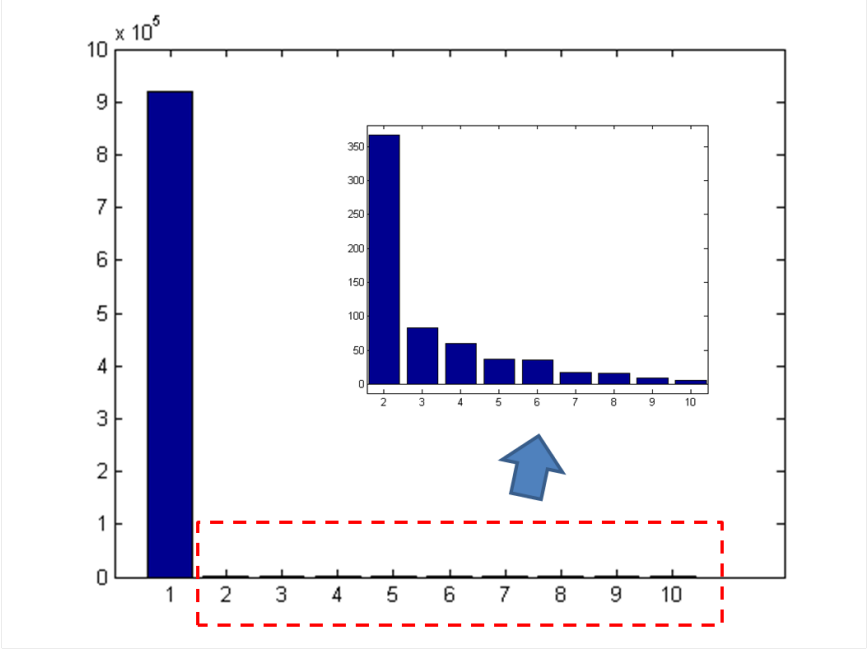

Figure 3.10: Hankel singular values of H_∞ controller

implementing H_∞ controller. The first part is model reduction similar to that in Chapter 2. The second one is to discretize the continuous model of reduced controller and implement it into dSpace.

The hankel singular values obtained through balanced realization is presented in Figure 3.10. The first state is dominant over the rest, however we should keep as many states as the hardware can afford. Among states from 2nd to 10th, 9th and 10th are relatively small and thus are removed also due to hardware limitation. The simplified controller produces almost the same closed loop performance as shown in Figure 3.11.

Now we have a 8th order controller obtained by H_∞ and then model reduction. The next step is to get a discrete time model of this controller in order to improve efficiency for practical implementation. For linear time invariant system, we have its transfer function which describes the dynamics of the system. However, controller is implemented in digital system which only takes in a sequence of discrete input signal and output correspondingly a sequence of discrete output signals. Similar to Laplace transform, *z-transform* is a way to

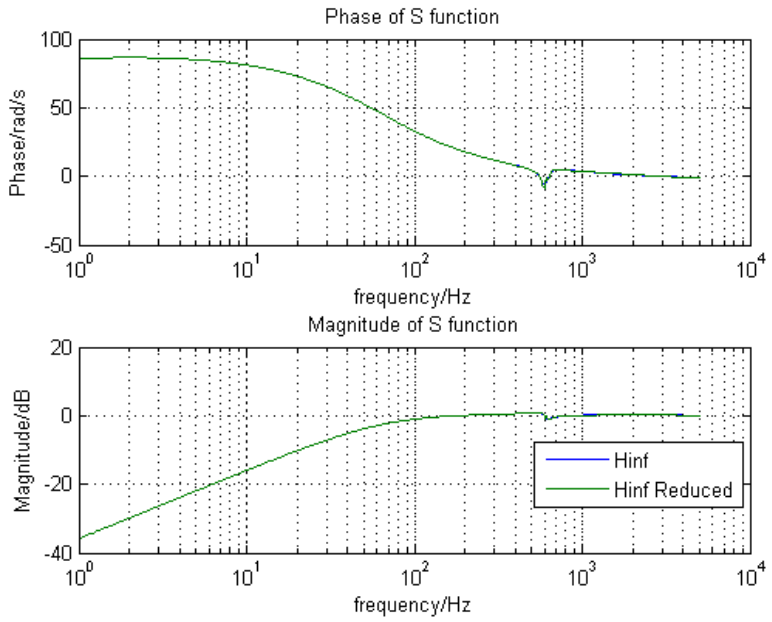

Figure 3.11: Sensitivity function S by simulation: original H_∞ controller V.S model-reduced controller

describe discrete time signal by complex variable given by

$$X(z) \equiv \sum_{n=-\infty}^{+\infty} x(n) z^{-n}, \tag{3.16}$$

where $x(n)$ is a discrete-time signal and z is a complex variable. Thus $X(z)$ is the representation of $x(n)$ in the complex plane. A digital filter with input x and output y is in the following form

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k). \tag{3.17}$$

Correspondingly, by z-transform, the input-output relation in complex domain is derived by

$$\begin{aligned} Y(z) &= -\sum_{k=1}^{N} a_k Y(z) z^{-k} + \sum_{k=0}^{M} b_k X(z) z^{-k}, \\ thus, H(z) &= \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}. \end{aligned} \tag{3.18}$$

Therefore, having $H(z)$, a digital filter is reconstructed by inverse z-transform. One can refer to Chapter 3 in [19] for further reading about *z-transform*.

Given a continuous-time filter, *Bilinear transformaiton* is implemented to transform it into a discrete-time representation. It is a first-order approximation of natural logarithm function that maps from z-plane to s-plane, or vice versa. Replace s in the continuous-time model with the approximation shown in shown in

$$\begin{aligned} s &= \frac{1}{T} ln(z) \\ &= \frac{2}{T}[\frac{z-1}{z+1} + \frac{1}{3}(\frac{z-1}{z+1})^3 + \frac{1}{5}(\frac{z-1}{z+1})^5 + \frac{1}{7}(\frac{z-1}{z+1})^7 + \cdots] \\ &\approx \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}, \end{aligned} \tag{3.19}$$

and then obtain a discrete-time model by first order approximation. For further reading about Bilinear transformation, one can refer to [20]

The process is further simplified by the matlab command *c2d* which directly converts a continuous model into a discrete one using the method "tustin". The return discretized controller is given by

$$K(z) = \frac{66.13z^8 - 289z^7 + 471.7z^6 - 243.7z^5 - 256.3z^4 + 482.7z^3 - 320.4z^2 + 102z - 12.95}{z^8 - 4.371z^7 + 7.014z^6 - 4.03z^5 - 0.883z^4 + 0.8039z^3 + 1.835z^2 - 1.885z + 0.5157}. \tag{3.20}$$

Then implement this discretized controller in dSpace. Obtain the sensitivity function S and transfer function T through experiment as shown Figure 3.12 and Figure 3.13 respectively. The experimental results matches well with simulation up to 500 Hz beyond which the time delay of dSpace takes the dominance. The actual ω_S is 68 Hz and ω_T is 74 Hz. Step responses and Tracking performance are also demonstrated in Figure 3.14 and Figure 3.15 respectively. The resolution of positioning achieved by the closed loop system is also shown in Figure 3.16. Both estimations are based on 10000 samples taken from each output respectively. The resolution is down to $0.0531\mu m$. This is a better result in terms of noise reduction compared

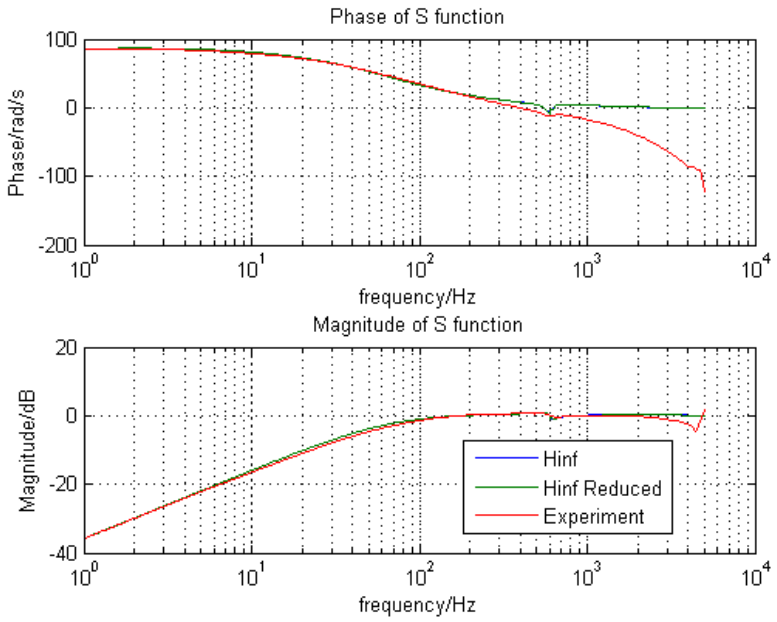

Figure 3.12: Sensitivity function S: experimental v.s simulation

to that given by open loop system as also shown in Fig 3.16. This is also one of the benefits for having feedback system rather that open loop ones, as discussed in [16].

The H_∞ controller renders a better closed loop performance compared with PID. From the design process discussed in this chapter, it is also shown that H_∞ control does not add tedious work load for control engineers while providing a better controller.

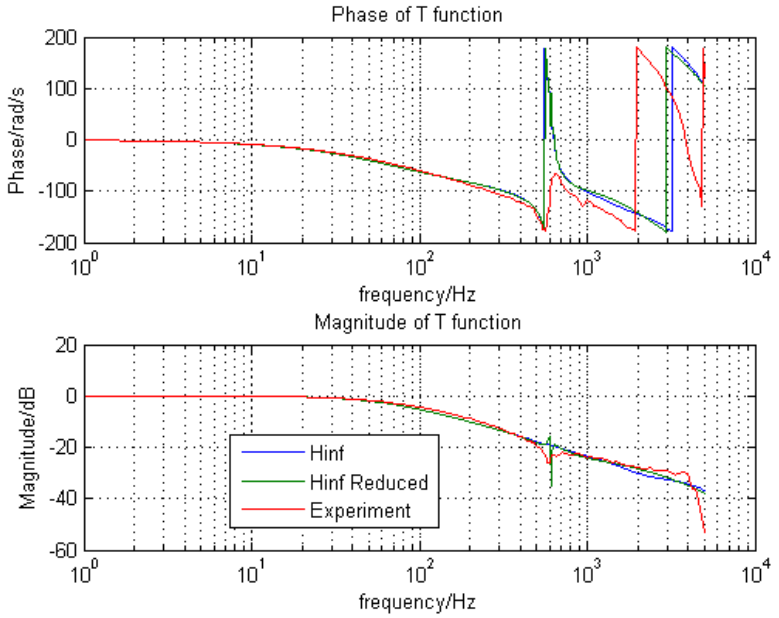

Figure 3.13: Transfer function T: experimental v.s simulation

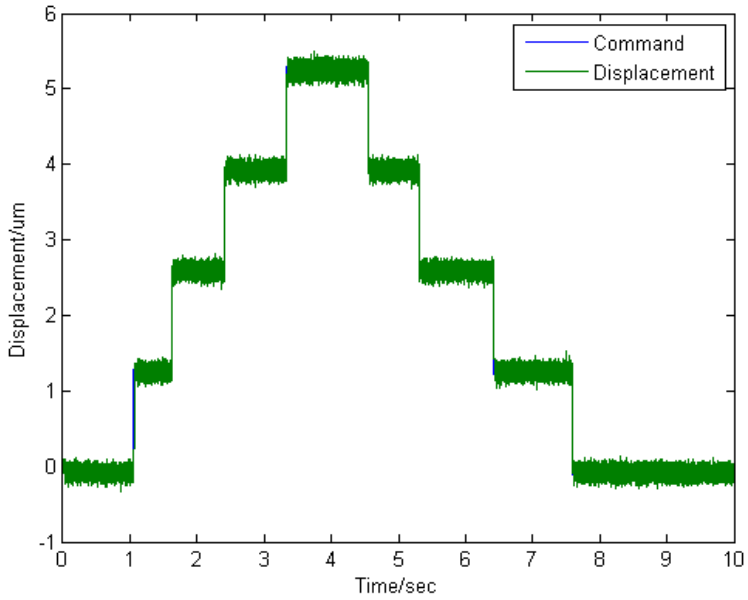

Figure 3.14: Step response of the closed loop system based on H_∞*controller*

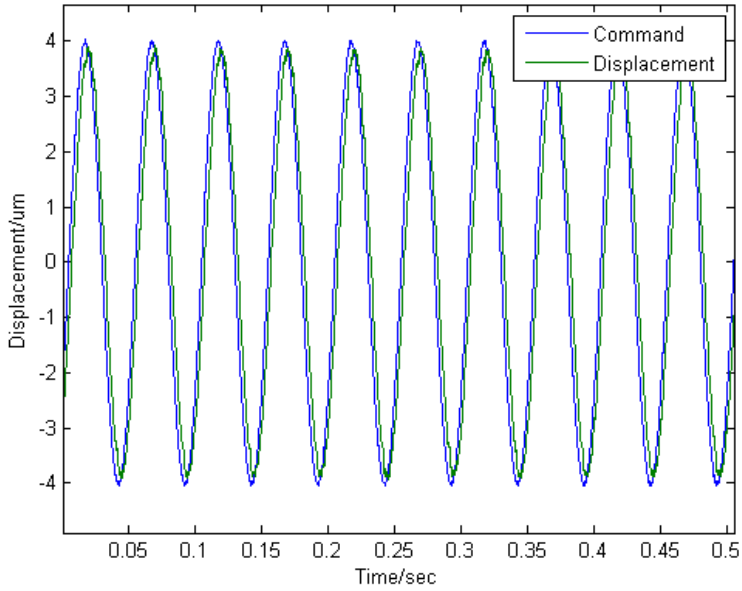

Figure 3.15: Tracking performance of the closed loop system based on H_∞ controller: tracking a 20 Hz sinusoidal command

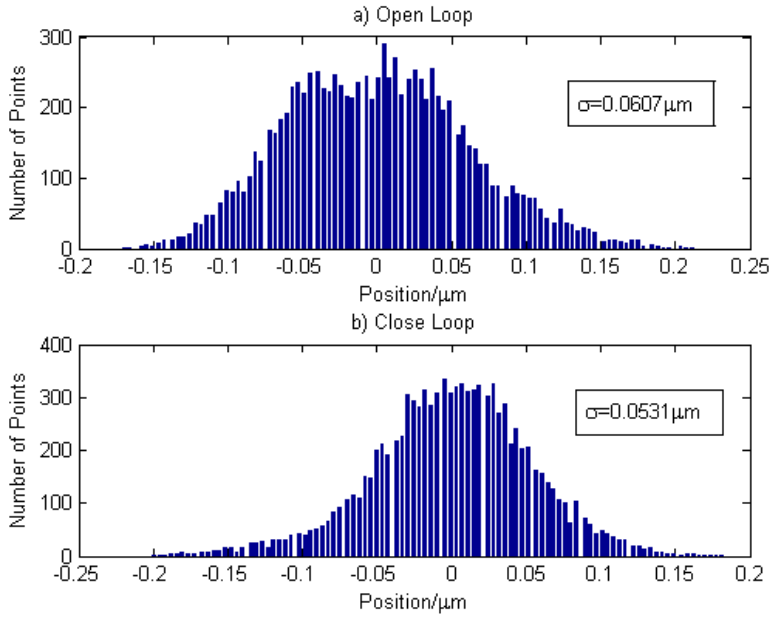

Figure 3.16: Histogram of measured comb drive positions around a steady state position: a) Open Loop; b) closed loop.

Chapter 4

Conclusion and Future Directions

A MEMS probing device, which is driven by an electrostatic comb drive is presented in this article. This MEMS device is a linear time invariant (LTI) system and has a mechanical dynamics that is similar to a typical underdamped second order system. Nonlinearity is introduced since the driving force F is proportional to V^2 where V is supply voltage for the comb drive. However this issue is solved by treating V^2 as the controller output. In order to capture the movement of the comb which changes linearly with corresponding comb capacitance, a universal capacitance readout IC is implemented and is able to produce output voltage that changes linearly with input capacitance. Thus by doing this, the displacement of comb drive around the operating point is estimated. However, the linear measurement of the sensor does have a limit. The sensor introduces a zero around 80 Hz in its transfer function and thus the measurement is linear up to around 80 Hz. This means the sensor reading is not reliable beyond that frequency. As a result, the target bandwidth ω_S is assigned to be less than 80 Hz, which also leaves room for improving other closed loop performances such as noise reduction. The model for XY stage, combined with the capacitance sensor, is identified and further simplified by model reduction method.

The main objective of implementing feedback control is to reduce effects from disturbance, attenuate noise, and improve model robustness. PID controllers are widely used by industry for their simplicity to implement. As shown in this article, empirical PID control does an impressive job even in controlling MEMS probing device which is difficult to control since it is sensitive to noise, disturbance, and often embed large model uncertainty. However, as noted in the Chapter 2, the implemented PID controller is an advanced one with

an additional pole included and often it is tedious to tune those parameter, K_P,K_I,K_D. In contrast, H_∞ control is based on convex optimization. What is needed is to design desired weight functions which defines the designed transfer function S, KS, and T. The resulting controller K is given by convex optimization with weight functions as constraints. H_∞ control is a highly automated design process with input of the process comprehensively describing desired closed loop performance. It is more advanced than PID control and remain flexible and easy to implement. The main goal of building up a feedback loop around MEMS device is tracking the reference position of XY stage along each axis. Since X and Y axe are decoupled, studying the position control along a single axis is enough. Therefore in this work, the feedback system is a single-input-single-output (SISO) one. The design on weighting functions for S, KS, and T are given in Chapter 3. The main idea behind it is to balance closed loop bandwidth and noise attenuation, stability and robustness since there are fundamental constraints for any feedback system. There is also a 80 Hz frequency limit imposed by the purchased capacitive sensor. The resulting controller is of 10th order and is however, beyond the reach of the dSpace control board. It is then necessary to implement model reduction method based on Balanced realization. The order of H_∞ controller is therefore reduced to 8th and the simplified controller is further discretized by Tustin method for the purpose of improving implementation efficiency. The closed loop performances are demonstrated in both frequency and time domain. From experimental results, the H_{infty} controller renders a feedback system which has a better closed loop bandwidth and a better robustness than the PID controller implemented in this work. Moreover, H_∞ control does not require tedious empirical tuning and provides a more comprehensive way of controller designing. This is important in many areas of controls, such as aerospace, nano-positioning since corresponding devices to control can not afford a Trial and Error approach. In conclusion, H_∞ control is an ideal way of controller design for nano-positioning application since it provides a comprehensive and computerized designing process for fashioning robust controllers. MEMS devices can be heavily effected by noise, disturbance and model uncertainty.

Therefore, the properties of H_∞ mentioned here are essential for these issues.

As the previous analysis tells, the bandwidth of the closed loop system is limited by the capacitive sensor used in experiment. In order to achieve the feedback control of scanning mode operating around the resonant frequency 600 Hz, a sensor with better bandwidth has to be brought in. Since the displacement control of comb drive along a single axis is achieved, the further work also includes designing H_∞ controller for a Multi-Input-Multi-Output (MIMO) system which has position references for each of X and Y axis as inputs and displacements along both axe as outputs. Since the displacements along the two axis are decoupled theoretically and the only changes in control design process are replacing scalar weight functions with matrix ones, it is expected that H_∞ control method can also achieve a feedback system for the 2-axis motion control of XY stage with high bandwidth, robustness, stability, and noise reduction.

References

[1] W.C.Tang, T.C.Nguyen, M.W.Judy, and R.T.Howe, "Laterally driven polysilicon resonant microstructures," *Sensors and Actuators*, vol. 20, pp. 25–32, 1989.

[2] B.Borovic, C.Hong, X.M.Zhang, A.Q.Liu, and F.L.Lewis, "Open vs. closed-loop control of the mems electrostatic comb drive," in *Proceedings of the 13th Mediterranean Conference on Control and Automation*, Limassol, Cyprus, June 2005.

[3] J. Dong and P. Ferriera, "Simultaneous actuation and displacement sensing for electrostatic drives," *Journal of Micromechanics and Microengineering*, vol. 18, Jan 2008.

[4] P.F.Indermuhle, V.P.Jaecklin, J.Brugger, C.Linder, N. Rooij, and M.Binggeli, "AFM imaging with an xy-micropositioner with integrated tip," *Sensors and Actuators*, pp. 562–565, 1995.

[5] J. Bryzek, A. Flannery, and D. Skurnik, "Integrating microelectromechanical systems with integrated circuits," *IEEE Instrumentation and Measurement Magazine*, vol. 7, pp. 51–59, June 2004.

[6] J. I. Seeger and B. E. Boser, "Charge control of parallel-plate, electrostatic actuators and the tip-in instability," *Journal of Microelectromechanical Systems*, vol. 12, no. 5, Oct 2003.

[7] K. P. Tee, S. S. Ge, and F. E. H. Tay, "Adaptive control of electrostatic microactuators adaptive control of electrostatic microactuators with bidirectional drive," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 2, pp. 340–352, 2009.

[8] K. L.Barton and A. G. Alleyne, "A cross-coupled iterative learning control design a cross-coupled iterative learning control design for precision motion control," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1218–1231, 2008.

[9] J. Dong, S. M.Salapaka, and P. M.Ferreira, "Robust mimo control of a parallel kinematics nano-positioner for high resolution high bandwidth tracking and repetitive tasks," in *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans, LA, Dec 2007.

[10] B. Koo, J. Dong, S. Salapaka, and P. M. Ferreira, "A 2 degree-of-freedom soi-mems translation a 2 degree-of-freedom soi-mems translation a 2 degree-of-freedom soi-mems translation stage with closed loop positioning," 2011.

[11] *MS3110 Datasheet*, Irvine Sensors Corporation, 3001 Redhill Avenue, Costa Mesa, California 92626.

[12] T. Soderstrom and P. Stoica, *System Identification*, ser. System and Control Engineering, M.J.Grimble, Ed. Prentice Hall, 1989.

[13] E. Levis, "Complex-curve fitting," *IRE Trans. on Automatic Control*, vol. AC-4, pp. 37–44, 1959.

[14] J. Dennis, J.E. and R. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, E. Cliffs, Ed. NJ: Prentice-Hall, 1983.

[15] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control-Analysis and Design*. Wiley, 1996.

[16] K. J. Astrom and T. Hagglund, *Advanced PID Control*. ISA-Instrumentation, Systems, and Automation Society, 2006.

[17] K. Glover and J. Doyle, "State-space formulae for all stabilizing controllers that satisfy an hinf norm bound and relations to risk sensitivity," *Systems and Control Letters*, vol. 11, pp. 167–172, 1988.

[18] K. G. P. K. Doyle, J.C. and B. Francis, "State-space solutions to standard h2 and hinf control problems," *IEEE Transactions on Automatic Control*, vol. 34, no. 8, pp. 831–847, August 1989.

[19] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applicaitons*, 3rd ed. Prentice Hall, inc, 1996.

[20] Y. Matsuno, *Bilinear Transformation Method*. Orlando, Florida 32887: Academic Press, Inc, 1984.